Seraphis Tx Protocol

https://github.com/UkoeHB/Seraphis

Part 1: Tx Protocols

Tx Protocol Structure ('Enote' Paradigm)

- Enote: an 'amount' that is 'owned' (aka output/Txo)
  - Ownership: exclusive power to assign a new owner
- Transaction Events: spend owned enotes, create new enotes
  - Inputs: spending old enotes
    - Membership: enotes exist in canonical enote set
    - Ownership: tx author has right to spend enotes
    - Unspent: old enotes have not been spent before
  - Outputs: new enotes with new owners
    - Amount Balance: sum(inputs) = sum(outputs) + fee

Need-to-know Design (Privacy)

- recipients
  - know: amount received, and when
  - don't know: sender
- senders
  - know: amount sent, when, [to who]
  - don't know: if/when sent money is spent
- observers
  - know: # inputs/outputs in a tx event, tx fees, ~ when tx constructed
  - don't know: amounts in any tx event (except fee), relationships between tx events, amounts owned by any user

Need-to-know Design (Privacy)

- recipients
  - know: amount received, and when
  - don't know: sender
- senders
  - know: amount sent, when, [to who]
  - don't know: if/when sent money is spent
- observers
  - know: # inputs/outputs in a tx event, tx fees, ~ when tx constructed
  - don't know: amounts in any tx event (except fee), relationships between tx events, amounts owned by any user

Problem: Membership Proofs

- how to:
  - show enote exists on-chain
  - don't show which enote it is
- solution 1: partial anonymity sets (ring signatures)
  - reference small subset, demonstrate membership in that subset
- solution 2: complete anonymity sets (ZCash-style oblivious Merkle proofs, ???)

Historical Protocols

- CryptoNote, RingCT, Triptych, Lelantus, Omniring, ...
- combined proofs: membership + ownership + unspent
- costs
  - complexity: satisfy multiple requirements at once
  - efficiency: more effort for ownership/unspent proofs

Part 2: Seraphis

Seraphis Tx Structure

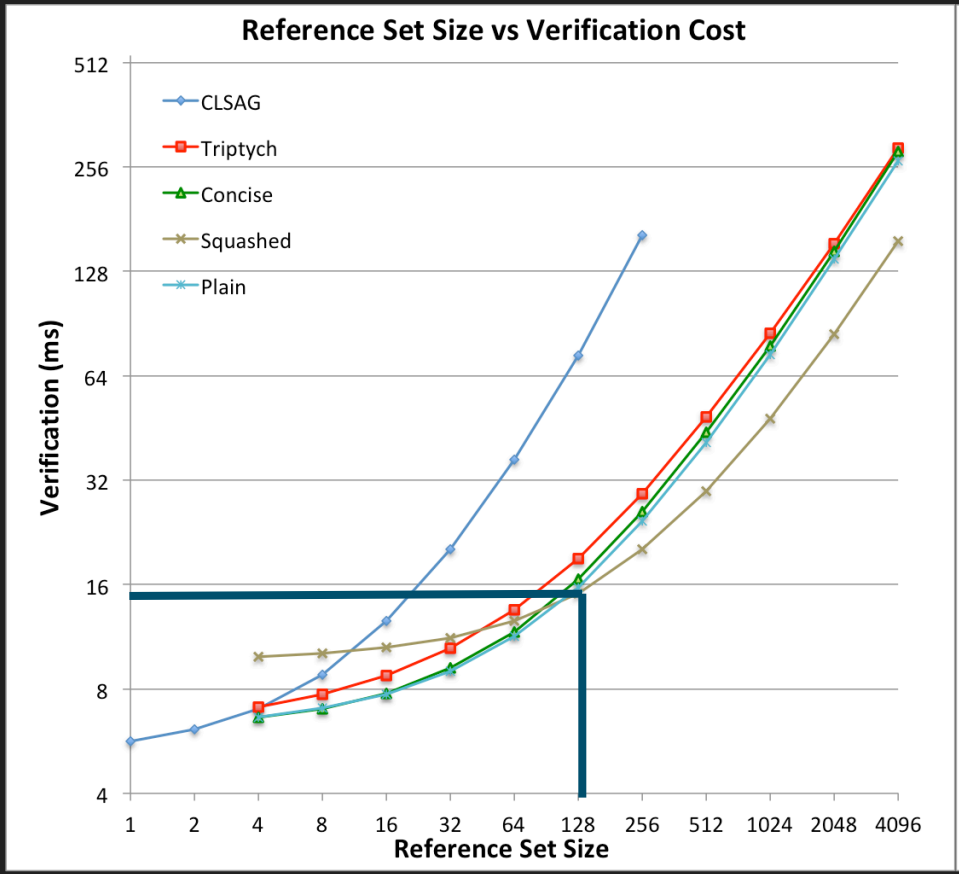
- Tx Inputs
  - Enote image: representation of an enote being spent
  - Membership proof: e.g. ring signature (membership only)
  - Ownership proof: enote owner authorizes spend (+ key image proof)
- Tx Outputs
  - Enote: new {amount, owner} message
  - Range proof: helps ensure input amounts == output amounts
- Balance Proof: input amounts == output amounts
- Misc.
  - Tx Fee, Memos

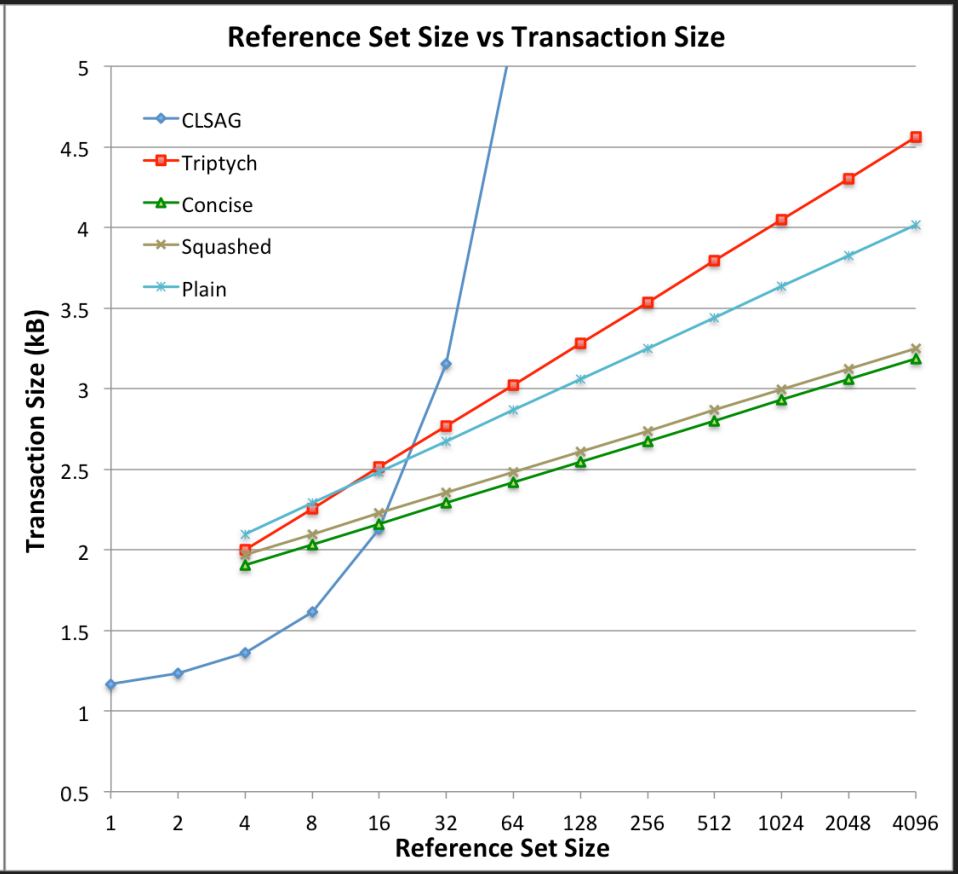
https://github.com/UkoeHB/monero/tree/seraphis_lib

Seraphis Tx Engineering

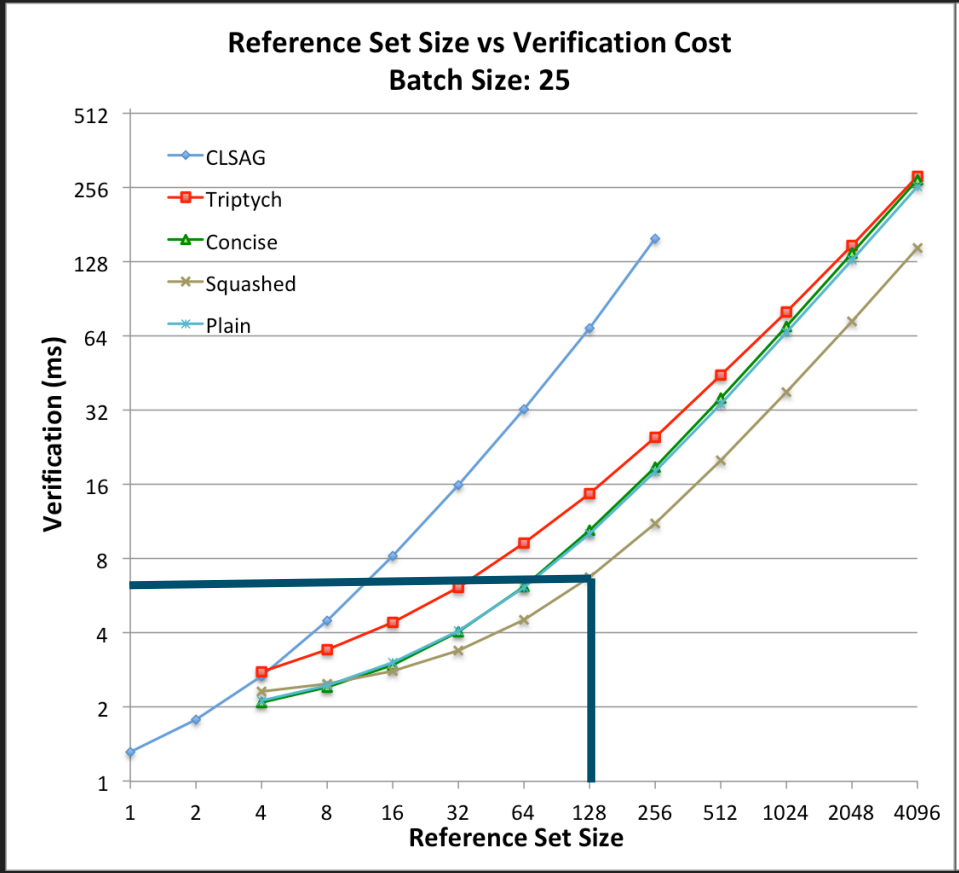
- Isolated Enote
- Delayed Membership Proofs (can be delegated)
  - multisig tx proposal for unconfirmed tx
  - simpler hw wallet implementation
- Tx Chaining
  - chain off tx proposal OR isolated enote
  - sign tx that spends enotes not in ledger yet (atomic swaps)
- Partial Funding
  - collaborate with many people to fund a tx

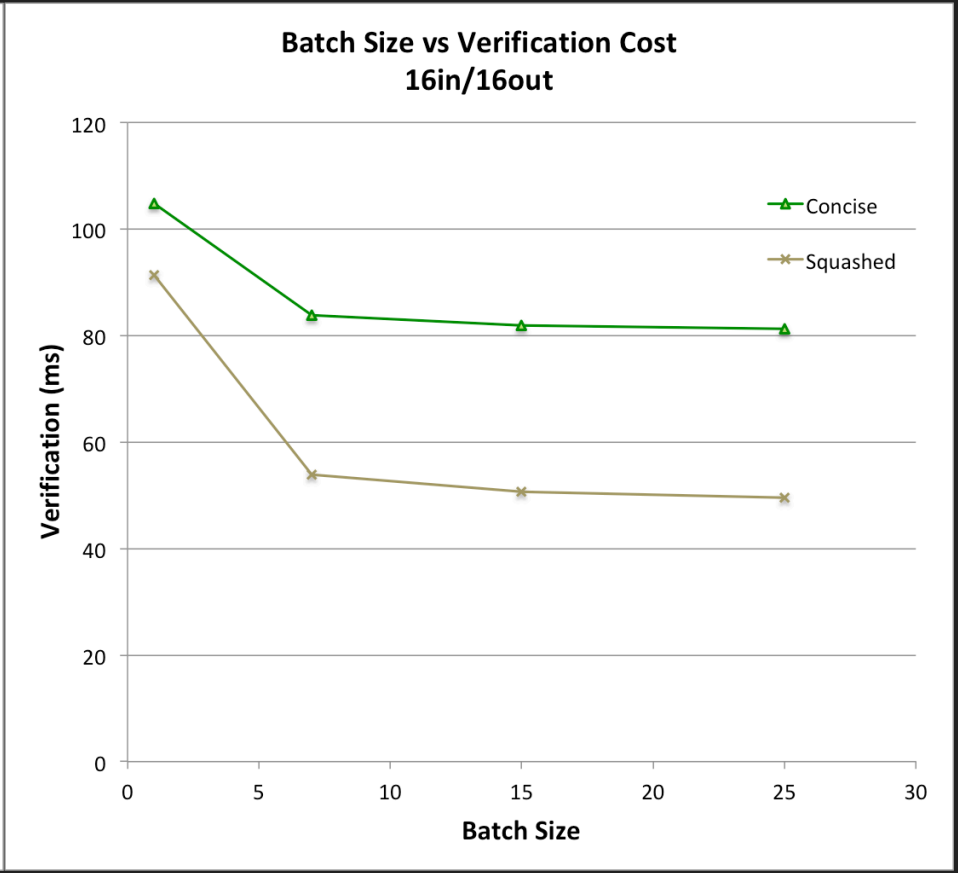
Seraphis Performance (1)





Seraphis Performance (2)





Part 3: Jamtis

Jamtis

- New Mnemonic: 16 words (Polyseed) with embedded restore height
- One Address Type: no more normal/subaddress/integrated
  - xmr1majob1977bw3ympyh2yxd7hjymrw8crc9kinodkm8d3wdu8jdhf3fkdpmgxfkbywbb9mdwkhkya4jtfnod5h7s49bfyji1936w19tyf396ypjo9n64runqjrxwp6k2s3phxwm6wrb5cob6c1ntrg2mugeocwdgnnr7u7bgknya9arksrj
- Recipient IDs (RIDs): manually validate an address
  - regne-hwbna-u21gh-b54no-8x36q

https://gist.github.com/tevador/50160d160d24cfc6c52ae02eb3d17024

Jamtis Wallet Tiers

- Master: view full balance, spend
  - normal wallet, hw wallet
- View All: view full balance
  - view-only wallet, hw/cold wallet hot interface
- View Received: view non-change enotes (amount, address index)
  - payment validation
- Find Received: pre-process enotes to speed up View All scanning
  - third-party scanning service, private scanning server
- Generate Address: make addresses with random indices
  - merchant point-of-sale

End